

# Logics of Allies and Enemies: A formal approach to the dynamics of social balance theory\*

Wiebe van der Hoek<sup>1</sup>, Louwe B. Kuijer<sup>1</sup> and Yi N. Wang<sup>2</sup>

<sup>1</sup>University of Liverpool

<sup>2</sup>Zhejiang University

{wiebe, louwe.kuijer}@liverpool.ac.uk, ynw@xixilogic.org,

## Abstract

Social balance theory studies the way different friendships and rivalries in a social network influence each other. Its main result is that, over time, social networks tend to become *balanced*. Here, we combine social balance theory with temporal logic. This yields an expressive language that can describe the plausible ways in which a social network might evolve over time.

## 1 Introduction

Social balance theory was initiated with Heider’s work on social psychology [1944; 1946; 1958], and later reinvented by Harary et al. using graph theory in [Harary, 1953; Cartwright and Harary, 1956; Harary *et al.*, 1965], an approach in which signed graphs represent social networks of agents, with *positive* signs for allies or friends and *negative* signs for enemies or antagonists. This has become a basic framework for studying positive and negative ties, and has since then become an active area in the field of social network analysis.

A social network is balanced if it meets certain structural conditions on its positive or negative ties between agents. For example, a triad of three agents that are all enemies of one another is considered unbalanced, since two of them have an incentive to make an alliance against the third. Over time, the ties in a social network tend to change in a way that makes the network more balanced. Empirical and theoretical studies of social balance can be found in [Newcomb, 1961; Doreian *et al.*, 1996; Hummon and Doreian, 2003; Wang and Thorngate, 2003; Antal *et al.*, 2006; Radicchi *et al.*, 2007; Kulakowski, 2007; Abell and Ludwig, 2009; Zheng *et al.*, 2015].

Here we study the process under which networks become more balanced from a different perspective, namely that of temporal logic. Specifically, we introduce the Logic of Allies and enemies (LAE), a variant of Computation Tree Logic (CTL) [Clarke and Emerson, 1981; Emerson and Clarke, 1982] that describes the behaviour of social networks under the assumption that they move towards balance “greedily”,

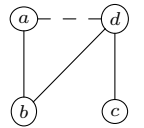
i.e., that change in relations between agents happens only if it makes the network more balanced.

LAE allows us to describe properties of networks such as, for example, “it is guaranteed that  $a$  and  $b$  eventually become friends”, “if  $a$  and  $c$  every become friends then they will remain friends forever, and  $a$  and  $b$  will forever be enemies” and “ $a$  and  $b$  will remain enemies until there are at least two agents  $x$  and  $y$  that are mutual friends of  $a$  and  $b$ ”. Our main results are that (1) it is possible for a social network to get stuck in a local maximum of balance, (2) the so-called *balance theorem* holds for LAE and (3) model checking and validity checking for LAE are PSPACE-complete.

The structure of the paper is as follows. We introduce the basic ideas of network balance in the next section. In Section 3 we introduce the syntax and semantics of LAE, together with a number of validities; we also show that the movement towards balance may terminate in a state that is not fully balanced by instead only *stable*. We study the computation complexity in Section 4 and conclude in Section 5.

## 2 Network Balance

A social network consists of a set of agents with pairwise ties that are positive (“friends”, “allies”, +), negative (“foes”, “enemies”, “hostile”, −) or neutral (“neither friends nor foes”, 0). It is customary to draw a network as a graph, with solid lines representing positive relations, dashed lines representing negative relations and the absence of a line representing neutral relations. For example, the diagram above represents a network where there is a friendship between  $a$  and  $b$ , between  $b$  and  $d$  and between  $c$  and  $d$ , an enmity between  $a$  and  $d$  and neutral relations between  $a$  and  $c$  and between  $b$  and  $c$ .



While every relation is between exactly two agents, the different relations do influence one another. The most famous of these influences is probably the saying that “the enemy of my enemy is my friend”, so if there is enmity between  $a$  and  $b$  and between  $b$  and  $c$ , then there should be a friendship between  $a$  and  $c$ . There are multiple equivalent ways to describe these influences; our description is based on balanced and unbalanced triads.<sup>1</sup> Modulo symmetry there are 10 different triads,

\*This is the author accepted version of a paper that was published in the proceedings of IJCAI2020 (<http://www.ijcai.org>)

<sup>1</sup>The term “triad” is commonly used in the field of balance theory for a group of three agents and their relations.

which are drawn in Figure 1.

We call a triad *balanced* if its edges reinforce each other. For example, in the triad  $---+$  of Figure 1(b), the  $+$  edge between  $a$  and  $c$  is reinforced because  $b$  is a mutual enemy, the  $-$  edge between  $a$  and  $b$  is reinforced because  $a$  is the friend of  $b$ 's enemy  $c$  and the  $-$  edge between  $b$  and  $c$  is reinforced because  $c$  is the friend of  $b$ 's enemy  $a$ . The two balanced triads are the aforementioned  $---+$  and the triad  $+++$ .

We call a triad *unbalanced* if all its edges weaken one another. For example, in the  $---$  triad of Figure 1(d) each pair of agents has reason to become friendly to one another against a common foe. The other unbalanced triad is  $++-$ .

There are also three triads where two of the edges are neither reinforced nor weakened, while a third edge experiences some pressure one way or the other. We call these triads *partially balanced*. For example, in the triad  $--0$  of Figure 1(f), there is no reason for the enmity between  $a$  and  $b$  or between  $b$  and  $c$  to end. But there is a reason for the neutrality between  $a$  and  $c$  to turn into a friendship, since they have  $b$  as a common foe.

Finally, the remaining three triads are *pressure-free*. In these triads  $000$ ,  $+00$ ,  $-00$ , there is no pressure on any of the edges to change.

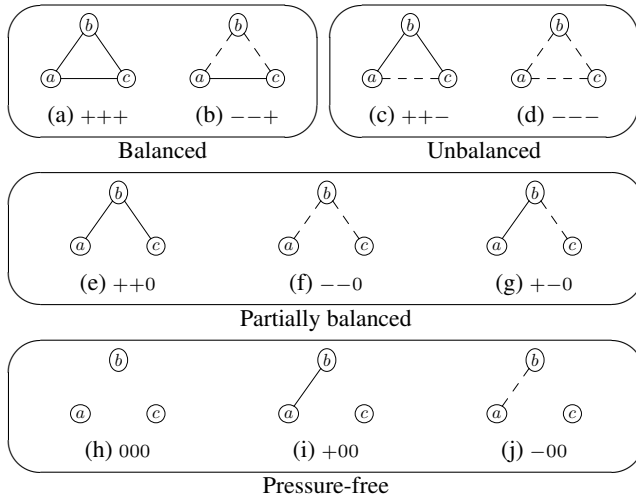


Figure 1: The ten different triad shapes

The different types of triads and the pressures they experience are summarized in the following table:

Degree of balance	Shape	Has a reason to change to ...
Balanced	+++	N/A
	---+	N/A
Unbalanced	++-	$-+-$ , $+-$ or $+++$
	---	$+-$ , $-+-$ or $---+$
Pressure-free	000	N/A
	+00	N/A
	-00	N/A
Partially balanced	++0	+++
	--0	---+
	+-0	++-

## 2.1 Scope

Balance theory models the influence that different relations in a social network have on one another. But the relations may be influenced by other factors as well. For example, "John may be the enemy of my enemy, but he punched me in the face yesterday so he is definitely not my friend" is a reasonable attitude, but does not follow from any of the relations in the network. Since such influences originate from factors that are not represented in the network, we refer to them as *outside influence*.

Predicting outside influence would require an accurate model of all human behavior, which seems rather unfeasible and is outside the scope of balance theory, and of this paper. We therefore do not fully model outside influence. The network dynamics that we represent here are therefore best seen as the *likely changes* in the social network *provided that there are no influences from outside the network*.

## 3 The Logic of Allies and Enemies

In this section we first introduce formal definitions of 3-signed social networks, stability scores and time evolution, based on the understanding and convention from the previous section. Then we introduce the syntax and semantics of the Logic of Allies and Enemies (LAE).

We assume that every social network is finite, so let a finite set  $AG$  be given. Furthermore, because there are no reasons for friendship or enmity unless there are at least three agents, we assume that  $|AG| \geq 3$ .

### 3.1 3-signed social networks

We define a social network to be a 3-signed undirected graph, with its vertices representing agents and edges representing ties between agents. The formal definition is given below.

**Definition 1** (social networks). A social network (network for short) is a function  $N : \{\{a, b\} \subseteq AG \mid a \neq b\} \rightarrow \{+, -, 0\}$  that assigns to each pair of different agents a positive (+), a negative (-), or a neutral (0) edge.

Note that the domain of  $N$  consists of unordered pairs of two different agents. We therefore have  $N(\{a, b\}) = N(\{b, a\})$ , by definition. We write  $N(a, b)$  for  $N(\{a, b\})$ . We say that a network is complete if it does not contain neutral edges, and that two agents are in the same connected component if they are connected by a path of non-neutral edges.

We first introduce the notion of *balance* coming from the literature. That is, a network is balanced if all of its triads are balanced.

**Definition 2** ((semi-)balance). A network  $N$  is balanced if for every distinct  $a, b, c \in AG$ , the triad  $abc$  is balanced in  $N$ . A network  $N$  is semi-balanced if for every distinct  $a, b, c \in AG$ , the triad  $abc$  is balanced or pressure-free in  $N$ .

The relation between the two concepts is characterized by the following proposition.

**Proposition 3.** A network  $N$  is semi-balanced if and only if all of its connected components are balanced.

A network is semi-balanced if and only if no agent has any reason to change any of its relations. However, it is possible

for two agents  $a$  and  $b$  to have both some reasons to become or remain friends, and some reasons to become or remain enemies. So the mere fact that an agent has a *prima facie* reason to change its relation does not mean that the agent has an overall reason to change its relation, since the reason for change may be outweighed by more reasons for the relation to remain the same. As a result, it is possible for a network to be non-balanced, while still not containing any overall reason for any relation to change. We therefore introduce the notion of *stability*: a network is unstable if there is at least one pair of agents that have more reasons to change their relationship than they have reasons for it to remain the same.

We determine the stability of an edge by computing its *attraction* and *repulsion*. We define the attraction between two agents to be the number of reasons for them to become or remain friends, and the repulsion the number of reasons to remain or become enemies. This means that the attraction between  $a$  and  $b$  is the number of mutual friends/foes, while the repulsion is the number of agents to which  $a$  and  $b$  have different non-neutral ties.

**Definition 4** (stability of edges). *Let  $N$  be a social network, and let  $a, b \in \text{AG}$ . The attraction, repulsion and stability score of  $(a, b)$ , denoted  $\text{attr}(a, b)$ ,  $\text{rep}(a, b)$  and  $\text{score}(a, b)$ , respectively, is given by*

$$\begin{aligned} \text{attr}(a, b) &= |\{c \mid N(a, c) = N(b, c) \text{ and } N(a, c) \in \{+, -\}\}| \\ \text{rep}(a, b) &= |\{c \mid N(a, c) = +, N(b, c) = -\}| + |\{c \mid N(a, c) = -, N(b, c) = +\}| \\ \text{score}(a, b) &= \begin{cases} \text{attr}(a, b) - \text{rep}(a, b), & \text{if } N(a, b) = +, \\ \text{rep}(a, b) - \text{attr}(a, b), & \text{if } N(a, b) = -, \\ -|\text{attr}(a, b) - \text{rep}(a, b)|, & \text{if } N(a, b) = 0. \end{cases} \end{aligned}$$

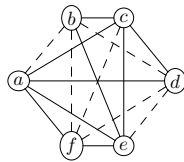
An edge  $ab$  is stable if  $\text{score}(a, b) \geq 0$ , and unstable otherwise. An unstable edge is improving if its attraction is greater than repulsion, and deteriorating if repulsion is greater than attraction.

Put in a different way, a positive edge is unstable if it is deteriorating, a negative edge is unstable if improving, and a neutral edge is unstable if either deteriorating or improving.

**Definition 5** (stability of networks). *A network is stable if every edge is stable, and unstable otherwise. The stability score of a network  $N$  is the sum of the stability scores of its edges. We write  $\text{score}(N)$  for the stability score of  $N$ .*

It is easy to see that every semi-balanced network is stable: if no edge has any reason to change then, in particular, no edge has more reasons to change than to remain the same. The converse does not hold, however.

For example, the network on the right is stable but not balanced. For every edge in this network the agents have a reason to change their relation, but also a countervailing reason to keep their current relation. E.g.,  $a$  and  $b$  might wish to become friends due to their common friendship with  $c$  and  $e$ , but they also wish to remain enemies due to the fact that  $d$  and  $f$  are enemies of  $b$  but friends of  $a$ .



**Proposition 6.** *The set of semi-balanced networks is a proper subset of the set of stable networks.*

If a network is unstable, one or more relations have more reasons to change than to remain the same. In such a situation, we expect one of these relations to change accordingly.

**Definition 7** (successors). *We say a network  $N_2$  is a successor of a network  $N_1$  if: (i)  $N_1$  is stable and  $N_1 = N_2$ , or (ii)  $N_1$  and  $N_2$  differ in exactly one edge  $ab$ , and for that edge it is either the case that  $ab$  is improving in  $N_1$  and  $N_2(a, b) = +$ , or that  $ab$  is deteriorating in  $N_1$  and  $N_2(a, b) = -$ . We write  $N_1 \rightsquigarrow N_2$  if  $N_2$  is a successor of  $N_1$ .*

**Proposition 8.** *Let  $N_1$  and  $N_2$  be two networks such that  $N_1 \rightsquigarrow N_2$ . Then,*

1. *If  $N_1$  is stable, then  $\text{score}(N_1) = \text{score}(N_2)$ ;*
2. *If a neutral edge of  $N_1$  becomes positive or negative in  $N_2$ , then  $\text{score}(N_1)$  can be greater than, equal to or smaller than  $\text{score}(N_2)$ ;*
3. *If a positive (resp. negative) edge of  $N_1$  becomes negative (resp. positive) in  $N_2$ , then  $\text{score}(N_1) < \text{score}(N_2)$ .*

**Definition 9** (time evolution). *Let  $N$  be a social network. The time evolution  $T(N)$  of  $N$  is the smallest graph  $(S, \rightsquigarrow)$  such that: (i)  $N \in S$ , and (ii) if  $N_1 \in S$  and  $N_2$  is a successor of  $N_1$ , then  $N_2 \in S$  and  $N_1 \rightsquigarrow N_2$ .*

The time evolution of  $N$  is not necessarily a tree, nor is it acyclic. However, by Prop. 8 the only cycles are self-loops, i.e., of the form  $N_1 \rightsquigarrow N_1$ , where  $N_1$  is stable. For along a *timeline* (i.e., a branch of the time evolution), either the stability scores of the involved networks increase or the networks contain less neutral edges (neutral edges can change to be positive or negative, but not vice versa). We can therefore abuse notation by saying that  $T(N)$  has a depth, namely the length of the longest cycle-free path.

**Proposition 10.** *The depth of  $T(N)$  is bounded from above by  $2 \cdot |\text{AG}|^5$ .*

*Proof.* Each reason for an edge to improve or deteriorate can be identified by a triad  $abc$ , where  $ab$  and  $bc$  influence  $ac$ . There are less than  $|\text{AG}|^3$  such triples, so the stability score must be between  $-|\text{AG}|^3$  and  $|\text{AG}|^3$ . Furthermore, there are at most  $|\text{AG}|^2$  edges with value 0. Since every non-reflexive transition requires either one edge to become non-neutral or the stability score to increase, the maximum cycle-free path length is at most  $2 \cdot |\text{AG}|^3 \cdot |\text{AG}|^2 = 2 \cdot |\text{AG}|^5$ .  $\square$

### 3.2 Syntax and semantics

We wish to reason about social networks and their evolution with a logical language. As  $T(N)$  strongly resembles the branching time models of CTL, we base the temporal connectives of our language on those of CTL. In addition to these temporal connectives we use a number of atoms that describe the ties. Because it is important for the network dynamics how many mutual friends/foes two agents have, we also use a quantifier  $\exists_{\geq n}x$  that allows us to describe how many agents satisfy a certain property.

**Definition 11** (languages). *The language  $\mathcal{L}$  of LAE is given by the following grammar:*

$$\varphi ::= Paa \mid Naa \mid \neg\varphi \mid (\varphi \rightarrow \varphi) \mid AX\varphi \mid A(\varphi U \varphi) \mid E(\varphi U \varphi) \mid \exists_{\geq n}x\varphi$$

where  $a, x \in \text{AG}$  and  $n \in \mathbb{N}$ . We use the Boolean operators  $\wedge, \vee, \top$  and  $\perp$  as well as the CTL operators  $EX, AF, EF, AG$  and  $EG$  as abbreviations as usual.  $Oab$  is used as an abbreviation for  $(\neg Pab \wedge \neg Nab)$ . With regard to the quantifier,  $\exists_{\geq n} x \varphi$  reads as “there are at least  $n$  agents  $x$  such that  $\varphi(x)$  is true.” We will make use of the following abbreviations:

$$\begin{aligned}\exists x \varphi &=_{\text{df}} \exists_{\geq 1} x \varphi \\ \forall x \varphi &=_{\text{df}} \neg \exists x \neg \varphi \\ \exists_{!n} x \varphi &=_{\text{df}} (\exists_{\geq n} x \varphi \wedge \neg \exists_{\geq n+1} x \varphi)\end{aligned}$$

For simplicity we treat  $Pab$  and  $Pba$  as the same formula, and similarly for the operators  $N$  and  $O$ .

Formally, the meaning of the formulas is determined by the satisfaction relation  $\models$ , which is defined as follows.

**Definition 12** (satisfaction). *Whether a network  $N$  satisfies (notation  $N \models$ ) a formula is determined inductively by the following (where  $\varphi$  and  $\psi$  are formulas, and  $a, b, x \in \text{AG}$ ):*

$$\begin{aligned}N \models Pab &\text{ iff } N(a, b) = + \\ N \models Nab &\text{ iff } N(a, b) = - \\ N \models \neg \varphi &\text{ iff not } N \models \varphi \\ N \models (\varphi \rightarrow \psi) &\text{ iff } N \models \varphi \text{ implies } N \models \psi \\ N \models AX\varphi &\text{ iff } \forall N' : \text{ if } N \rightsquigarrow N' \text{ then } N' \models \varphi \\ N_0 \models A(\varphi U \psi) &\text{ iff } \forall N_1, \dots, \text{ if } N_0 \rightsquigarrow N_1 \rightsquigarrow \dots \text{ then} \\ &\quad \exists i : N_i \models \psi \text{ and } \forall j < i, N_j \models \varphi \\ N_0 \models E(\varphi U \psi) &\text{ iff } \exists N_1, \dots \text{ such that } N_0 \rightsquigarrow N_1 \rightsquigarrow \dots \\ &\quad \exists i : N_i \models \psi \text{ and } \forall j < i, N_j \models \varphi \\ N \models \exists_{\geq n} x \varphi &\text{ iff there are distinct } a_1, \dots, a_n \in \text{AG} \\ &\quad \text{such that } N \models \varphi\left[\frac{a_1}{x}\right] \wedge \dots \wedge \varphi\left[\frac{a_n}{x}\right],\end{aligned}$$

where  $\varphi\left[\frac{a_i}{x}\right]$  (for all  $i = 1, \dots, n$ ) is the formula achieved by substituting all occurrences of  $x$  in  $\varphi$  to  $a_i$ .

A formula is satisfiable if there is a network that satisfies it, unsatisfiable if no network satisfies it, and valid if all networks satisfy it.

Our logic is strong enough to express *stability* of edges and networks and *balance* of triads and networks:

$$\begin{aligned}\text{stable}(a, b) &=_{\text{df}} (Pab \wedge AX Pab) \vee (Nab \wedge AX Nab) \vee (Oab \wedge AX Oab) \\ \text{stable} &=_{\text{df}} \forall xy \text{ stable}(x, y) \\ \text{noneutral}(a, b, c) &=_{\text{df}} \neg(Oab \vee Obc \vee Oac) \\ \text{balanced}(a, b, c) &=_{\text{df}} (Pab \wedge Pbc \wedge Pca) \vee (Pab \wedge Nbc \wedge Nca) \\ &\quad \vee (Nab \wedge Pbc \wedge Nca) \vee (Nab \wedge Nbc \wedge Pca) \\ \text{balanced} &=_{\text{df}} \forall xyz (\text{noneutral}(x, y, z) \rightarrow \text{balanced}(x, y, z))\end{aligned}$$

where  $\text{stable}(a, b)$  intuitively says that the edge  $ab$  is stable,  $\text{stable}$  says that the current network is stable ( $N \models \text{stable}$  iff  $N$  is stable),  $\text{noneutral}(a, b, c)$  says that  $abc$  contains no neutral edges,  $\text{balanced}(a, b, c)$  says that  $abc$  is a balanced triad (positive 3-cycle), and  $\text{balanced}$  says that the current network is balanced ( $N \models \text{balanced}$  iff  $N$  is balanced). Semi-balance can also be expressed in LAE, but not in a straightforward way. We explain how it can be expressed in Section 3.3. To further illustrate the expressive power of LAE, we list some validities below.

**Proposition 13.** *The following formulas are valid, where  $a, b, c$  and  $d$  are assumed to be distinct:*

1. (One is neither a friend nor an enemy of oneself)  $Oaa$
2.  $Pab \rightarrow \exists x Pax$ ,  $Nab \rightarrow \exists x Nax$  and  $Oab \rightarrow \exists_{\geq 2} x Oax$

3. (Stable networks don't evolve)  $\text{stable} \rightarrow (\varphi \rightarrow AX\varphi)$
4. (At most one sign is changed in one step)  $(Pab \wedge Pcd) \rightarrow AX(Pab \vee Pcd)$
5. (A network becomes stable eventually)  $AF\text{stable}$
6. (A network is stable after  $2 \cdot |\text{AG}|^5$  steps)  $AX^{2 \cdot |\text{AG}|^5} \text{stable}$
7. (Balance implies stability)  $\text{balanced} \rightarrow \text{stable}$
8. (There are  $|\text{AG}|$  agents)  $\exists_{\geq |\text{AG}|} x \varphi \leftrightarrow \forall x \varphi$

*Proof.* Statements 5 and 6 follow from the fact that the depth of  $T(N)$  is bounded by  $2 \cdot |\text{AG}|^5$ . The remaining statements follow easily from the definitions.  $\square$

### 3.3 Balance theorem revisited

Let us start by formally defining what we mean by the network being divisible into cliques.

**Definition 14.** A clique division of a network  $N$  is a partition  $V_1, \dots, V_k$  of  $\text{AG}$  such that (i) for all  $i = 1, \dots, k$  and all  $a, b \in V_i$ ,  $N(ab) = +$  and (ii) for all  $i, j$  such that  $1 \leq i < j \leq k$ , either  $N(a, b) = -$  for all  $a \in V_i$  and  $b \in V_j$  or  $N(a, b) = 0$  for all  $a \in V_i, b \in V_j$ .

A clique division is semi-bipartite if for every  $i$  there is at most one  $j$  such that for  $a \in V_i$  and  $b \in V_j$ ,  $N(a, b) = -$ .

A network is (semi-bipartite) clique divisible if it has a (semi-bipartite) clique division.

The property of being clique divisible, and of semi-bipartite clique divisible, can be expressed in LAE as follows:

$$\begin{aligned}\text{clique} &=_{\text{df}} \forall xy (Pxy \rightarrow (\forall z (Nxz \rightarrow Nyz) \wedge \neg \exists_{\geq 2} z (Pxz \wedge Oyz))) \\ \text{sbclique} &=_{\text{df}} \text{clique} \wedge \forall xyz \neg (Nxy \wedge Nyz \wedge \bar{N}xz)\end{aligned}$$

The definition of clique is not entirely straightforward, but the following proposition shows it is a faithful definition.

**Proposition 15.** *A network  $N$  is clique divisible iff  $N \models \text{clique}$ .  $N$  is semi-bipartite clique divisible iff  $N \models \text{sbclique}$ .*

*Proof.* A network is clique divisible iff for every  $a, b$  and  $c$ , if  $N(a, b) = +$  then  $N(a, c) = N(b, c)$ . We show that the formula  $\text{clique}$  guarantees exactly this property.

Take any two agents  $a$  and  $b$  such that  $N(a, b) = +$  and let  $c$  be a third agent. Suppose that  $N(a, c) \neq N(b, c)$ . Then we distinguish two possibilities: (i) one of  $a, b$  is an enemy of  $c$  and the other is not or (ii) one of  $a, b$  is a friend of  $c$  while the other is neutral. Without loss of generality, the negative relation in case (i) and the positive relation in case (ii) is between  $a$  and  $c$ . Then, in case (i), we have  $N \models Pab$  and  $N \not\models Nac \rightarrow Nbc$ , so  $N \not\models \text{clique}$ . In case (ii), we have  $N \models Pab$ ,  $N \models Pac \wedge Obc$  and  $N \models Pab \wedge Obb$ , so there are at least two witnesses for  $\exists_{\geq 2} z (Paz \wedge Obz)$ . Again,  $N \not\models \text{clique}$ .

Similar reasoning shows that if  $a$  and  $b$  have the same relation with every  $c$ , then  $\text{clique}$  holds. So  $\text{clique}$  holds if and only if  $N$  is clique divisible. Furthermore, a clique division is semi-bipartite if every clique is hostile to at most one other clique, so if no three cliques are mutually hostile. It is easy to see that this is the case iff  $\text{sbclique}$  holds.  $\square$

In Section 3.2 we claimed that the property of being semi-balance can be expressed in LAE. Here, we define a formula semi-balance that expresses this property. The definition

of semi-balanced is not entirely straightforward. The problem lies in the fact that  $Oab$  may hold because  $N(a, b) = 0$ , or it may hold because  $a = b$ . So while a semi-balanced network does not contain any triads of the form  $++0$ , it may contain agents  $x, y$  and  $z$  such that  $Pxz \wedge Pyz \wedge Oxy$ , if  $x = y$ . In the formula **semi-balanced** we have to take care of this special case.

$$\begin{aligned} \text{semi-balanced}(a, b) &=_{\text{df}} \forall x (((Pax \wedge Nxb) \rightarrow Nab) \wedge \\ &\quad ((Nax \wedge Pxb) \rightarrow Nab) \wedge \\ &\quad ((Pax \wedge Pxb) \rightarrow (Pab \vee (Oab \wedge AXOab))) \wedge \\ &\quad ((Nax \wedge Nxb) \rightarrow (Pab \vee (Oab \wedge AXOab)))) \\ \text{semi-balanced} &=_{\text{df}} \forall y \forall z \text{ semi-balanced}(y, z). \end{aligned}$$

**Proposition 16.** *A network  $N$  is semi-balanced if and only if  $N \models \text{semi-balanced}$ .*

*Proof.* Suppose that  $N$  is semi-balanced, and that  $N \models Pax \wedge Pxb$ . Then there are two possibilities: either  $a = b$  or  $abx$  is a semi-balanced triad with at least two positive edges. In the first case  $N \models Oab \wedge AXOab$ , since an agent is neutral to itself. In the second case,  $abx$  must be of the form  $+++$ , so  $N \models Pab$ . In either case,  $N \models (Pax \wedge Pxb) \rightarrow (Pab \vee (Oab \wedge AXOab))$ . The antecedents of the other three implications are false, so they are trivially satisfied. It can similarly be shown that if  $N \models Pax \wedge Nxb$ ,  $N \models Nax \wedge Pab$  or  $N \models Nax \wedge Nxb$  all four implications hold, so  $N \models \text{semi-balanced}$ .

Suppose then that  $N$  contains a triad  $abx$  such that  $N \models Pax \wedge Pxb \wedge Oab$ . Because  $a$  and  $b$  have a mutual friend  $x$ , the edge  $ab$  has an attraction of at least 1. There are two possibilities. Firstly, the edge  $ab$  may be improving. In that case,  $N \not\models AXOab$ , and therefore  $N \not\models (Pax \wedge Pxb) \rightarrow (Pab \vee (Oab \wedge AXOab))$ . Secondly, the attraction may be counterbalanced by a repulsion due to an anti-mutual relation with  $x'$ . In that case, we have either  $N \not\models (Pax' \wedge Nxb) \rightarrow Nab$  or  $N \not\models (Nax' \wedge Pxb) \rightarrow Nab$ . In either case,  $N \not\models \text{semi-balanced}$ . For all other unbalanced or partially balanced triads  $abx$  it can similarly be shown that  $N \not\models \text{semi-balanced}$ .  $\square$

Now that we have the formulas **semi-balanced** and **sbclique**, it is quite easy to formulate the Balance Theorem.

**Theorem 17** (balance).  $\models \text{semi-balanced} \leftrightarrow \text{sbclique}$ .

*Proof.* Suppose  $N \models \text{sbclique}$ . By Proposition 15 this implies that  $N$  is semi-bipartite clique divisible. Now, take any triad  $abc$  in  $N$ . If  $a, b$  and  $c$  are members of the same clique, then  $abc$  is of the form  $+++$ . If  $a$  and  $b$  are members of the same clique and  $c$  is a member of a different clique, then  $abc$  is of the form  $+-$ , if the cliques are hostile to each other, or  $+00$ , if the cliques are neutral to each other. If  $a, b$  and  $c$  are all members of different cliques, then due to the fact that  $N$  is semi-bipartite, at most two of these cliques can be hostile to one another. So  $abc$  is of the form  $-00$  or  $000$ . Each of these possible forms for  $abc$  is either balanced or pressure-free, so  $N$  is semi-balanced. By Proposition 16 this implies that  $N \models \text{semi-balanced}$ .

Suppose  $N \models \text{semi-balanced}$ . By Proposition 16,  $N$  is semi-balanced. Let  $V_1, \dots, V_k$  be the partition such that  $a, b$

are in the same part if and only if there is a path of positive relations from  $a$  to  $b$ . Because  $N$  is semi-balanced, positive relations are transitive. So every  $a, b \in V_i$  have positive relations. It follows that  $V_1, \dots, V_k$  is a clique division. Now, suppose towards a contradiction that there are cliques  $V_i, V_j, V_l$  that have negative relations with one another. Then for  $a \in V_i, b \in V_j$  and  $c \in V_l$ , the triad  $abc$  is of the form  $---$ . This contradicts  $N$  being semi-balanced. The clique division  $V_1, \dots, V_k$  is therefore semi-bipartite. By Proposition 15, this implies that  $N \models \text{sbclique}$ .  $\square$

## 4 Computational Complexity

Now that we have defined our logic of allies and enemies, we address the complexity of model checking and validity for this logic. Formally, the model checking problem is to determine, given a network  $N$  and a formula  $\varphi$ , whether  $N \models \varphi$ . The validity problem is to determine, given a formula  $\varphi$ , whether  $\models \varphi$ . We show that both the model checking problem and the validity problem are PSPACE-complete.

Before we can determine the complexity of either decision problem, we first need to define a measure for the input, and the complexity is then defined relative to this measure.

**Definition 18.** *The size of a formula  $\varphi$ , denoted  $|\varphi|$  is given recursively by*

$$\begin{aligned} |Pab| &= |Nab| = 1, & |\neg\varphi| &= |AX\varphi| = |\varphi| + 1, \\ |\varphi \rightarrow \varphi'| &= |A(\varphi U \varphi')| = |E(\varphi U \varphi')| = |\varphi| + |\varphi'| + 1, \\ |\exists_{\geq n} x \varphi| &= |\varphi| + n. \end{aligned}$$

The clause  $|\exists_{\geq n} x \varphi| = |\varphi| + n$  means that we assume that  $n$  is represented in unary. This is not critical for our results: all complexity results presented in this paper would still be true if we used a binary or decimal representation of  $n$ , with corresponding size measure  $|\exists_{\geq n} x \varphi| = |\varphi| + 1 + \log n$ .

The difficulty of the model checking and validity problems depends on AG as well as  $\varphi$ . This suggests two possible ways to define the input of the two problems. If we consider AG to be part of the input, then the input size for both problems is  $|\varphi| + |AG|$ . If we consider AG to be fixed, and therefore not part of the input, then the input size is  $|\varphi|$ . Fortunately, this distinction turns out not to matter: regardless of whether we consider AG to be part of the input, the model checking and validity problems are PSPACE-complete with respect to the relevant input size. In fact, our proofs in this section apply to either case.

### 4.1 Model checking in LAE: PSPACE-complete

We show that the complexity of the decision problem is PSPACE-complete. We start by proving hardness.

**Lemma 19.** *Model checking for LAE is PSPACE-hard.*

*Proof.* We use a reduction from the truth/satisfiability problem of quantified Boolean formulas (QBFSAT) which is known to be PSPACE-complete [Stockmeyer and Meyer, 1973]. Take any QBF instance  $Q_1 p_1 \dots Q_n p_n \psi(p_1, \dots, p_n)$ , where every  $Q_i$  is either  $\exists$  or  $\forall$ . Let  $N$  be the network such that  $N(ab) = +$  and  $N(x, y) = 0$  for  $\{x, y\} \neq \{a, b\}$ . We can now simulate the choosing of true/false of QBFSAT by

choosing either  $b$  (for truth) or any other agent (for false) using the quantifier of LAE. Note that we can recognize the choice because we have  $N \models Pab$  and  $N \models \neg Pac$  for any  $c \neq b$ . We have  $N \models Q_1x_1 \cdots Q_nx_n\psi(Pax_1, \dots, Pax_n)$  iff  $\models Q_1p_1 \cdots Q_np_n\psi(p_1, \dots, p_n)$ . The PSPACE-hardness of LAE model checking now follows immediately from the PSPACE-hardness of QBFSAT.  $\square$

**Example 20.** Consider the QBF instance  $\forall p \exists q (p \leftrightarrow q)$ , which is valid. The translation to LAE is  $\forall x \exists y (Pax \leftrightarrow Pay)$ . For every  $x \in AG$ , choose  $y \in AG$  in the following way: if  $x = b$ , then choose  $y = b$ , otherwise choose  $y = a$ . Then we have  $N \models Pax \leftrightarrow Pay$ . It follows that  $N \models \forall x \exists y (Pax \leftrightarrow Pay)$ . Consider then the QBF instance  $\exists p \forall q (p \leftrightarrow q)$  which is not valid. The translation to LAE is  $\exists x \forall y (Pax \leftrightarrow Pay)$ . Now, choose  $y$  in the following way: if  $x = b$ , then  $y = a$ , otherwise  $y = b$ . Then  $N \not\models Pax \leftrightarrow Pay$ . It follows that  $N \not\models \exists x \forall y (Pax \leftrightarrow Pay)$ .

Note that the reduction requires only 2 agents. This implies that model checking is PSPACE-hard regardless of the choice of AG. Left to show is that the model checking problem of LAE can be solved in polynomial space.

**Lemma 21.** Model checking for LAE is in PSPACE.

*Proof.* The temporal operators of LAE are those of CTL. The standard model checking algorithms for CTL [Clarke and Emerson, 1981] can therefore, with minor modifications, be used here. The algorithm is based on a case distinction regarding the main connective of  $\varphi$ . We present only two of these cases here, those of  $\exists_{\geq n}x\psi$  and  $A(\psi U \chi)$ , in Algorithms 1 and 2, respectively. In these cases, as in most other cases,

**Algorithm 1** Model checking algorithm for the case  $\exists_{\geq n}x\psi$

1: initialize $k = 0$	7: if $k \geq n$ then
2: for all $a \in V$ do	8: return true
3: if $N \models \psi[x/a]$ then	9: else
4: $k++$	10: return false
5: end if	11: end if
6: end for	

**Algorithm 2** Model checking algorithm for the case  $A(\psi U \chi)$

1: if $N \models \chi$ then	10: initialize $x = \text{true}$
2: return true	11: for all $N' \in \mathcal{N}$ do
3: else if $N \not\models \psi$ then	12: if $N' \not\models A(\psi U \chi)$
4: return false	13: then
5: else if $N \models \psi$ then	14: $x = \text{false}$
6: compute $\mathcal{N} = \{N' \mid N \rightsquigarrow N'\}$	15: end if
7: if $\mathcal{N} = \{N\}$ then	16: end for
8: return false	17: return $x$
9: else	18: end if

the algorithm recursively calls itself. In each case, however, the recursive calls are either for a strictly smaller formula or for a different network  $N'$  that occurs strictly deeper in the tree  $T(N)$ . Since  $T(N)$  is of finite depth and has no cycles except in leaf nodes, it follows that the algorithm terminates.

The usual algorithm for CTL runs in polynomial time, but that is with respect to  $|\varphi|$  and the size of the computation tree. In our case, this computation tree is  $T(N)$ , which is in general exponentially large with respect to  $|AG|$ . Fortunately, we do not need to keep all of  $T(N)$  in memory at once: we can free all memory needed to compute whether  $N' \models A(\psi U \chi)$  before we begin to compute  $N'' \models A(\psi U \chi)$ . In effect, this allows us to search  $T(N)$  in a depth-first way using an amount of memory polynomial in the depth of  $T(N)$ . As shown in Proposition 10, this depth is polynomial in  $|AG|$ . As such, model checking for LAE is in PSPACE.  $\square$

**Theorem 22.** Model checking for LAE is PSPACE-complete.

## 4.2 Validity checking in LAE: PSPACE-complete

We begin with PSPACE-hardness, for which we give, as usual, a reduction from QBFSAT.

**Lemma 23.** Validity checking for LAE is PSPACE-hard.

*Proof.* We use a formula  $\gamma$  to characterize models in which  $a$  and  $b$  are the only agents to have any non-zero edges to anyone else, and the relation between  $a$  and  $b$  is positive. I.e.,

$$\xi = Pab \wedge \forall x (\exists y (Pxy \vee Nxy) \rightarrow (Pxa \vee Pxb)) \wedge \neg \exists_{\geq 2} x Pax \wedge \neg \exists_{\geq 2} x Pbx.$$

Networks satisfying  $\xi$  are exactly of the form that was used in Lemma 19. If  $N \models \xi$ , then  $\models Q_1p_1 \cdots Q_np_n\psi(p_1, \dots, p_n)$  iff  $N \models Q_1x_1 \cdots Q_nx_n\psi(Pax_1, \dots, Pax_n)$ . It follows that  $\models Q_1p_1 \cdots Q_np_n\psi(p_1, \dots, p_n)$  iff  $\models \xi \rightarrow (Q_1x_1 \cdots Q_nx_n\psi(Pax_1, \dots, Pax_n))$ . This reduction does not depend on  $|AG|$ , so the validity problem of LAE is PSPACE-hard regardless of whether AG is a parameter.  $\square$

Left to show is PSPACE membership. This too can be shown using the result for model checking.

**Lemma 24.** Validity checking for LAE is in PSPACE.

*Proof.* For given AG there are finitely many different networks. This means that we can check whether  $\varphi$  is valid in LAE, using an exhaustive search. That is, for every network  $N$  we check whether  $N \models \varphi$ . Since model checking can be done in polynomial space (w.r.t.  $|N| + |\varphi|$ ) and we only need to keep one network in memory at a time, validity checking can be done in polynomial space with respect to  $|N| + |\varphi| = |AG| + |\varphi|$ . If we consider AG to be constant, this means that validity checking is in PSPACE with respect to  $|\varphi|$ . If we consider  $|AG|$  to be a parameter, then the problem is in PSPACE with respect to  $|AG| + |\varphi|$ . In either case, it is in PSPACE with respect to the relevant input size.  $\square$

The PSPACE-completeness of LAE follows immediately.

**Theorem 25.** Validity checking for LAE is PSPACE-complete.

## 5 Conclusion

We introduced a logic of allies and enemies (LAE), which combines social balance theory with temporal logic. LAE can be used to describe the likely evolution over time of relations in a social network.

An important concept from social balance theory is that of *balance*. We showed that the balance theorem can be formulated and proven in LAE. Furthermore, we showed that, in addition to *balance*, the weaker concept *stability* is important for understanding the behaviour of social networks. Finally, we showed that both model checking and validity checking for LAE are PSPACE-complete.

## References

- [Abell and Ludwig, 2009] Peter Abell and Mark Ludwig. Structural balance: A dynamic perspective. *The Journal of Mathematical Sociology*, 33(2):129–155, 2009.
- [Antal *et al.*, 2006] Tibor Antal, Paul L. Krapivsky, and Sidney Redner. Social balance on networks: The dynamics of friendship and enmity. *Physica D: Nonlinear Phenomena*, 224(1-2):130–136, December 2006.
- [Cartwright and Harary, 1956] Dorwin Cartwright and Frank Harary. Structure balance: A generalization of Heider’s theory. *Psychological Review*, 63(5):277–293, September 1956.
- [Clarke and Emerson, 1981] Edmund M. Clarke and E. Allen Emerson. Design and synthesis of synchronization skeletons using branching time temporal logic. In Dexter Kozen, editor, *Logics of Programs*, pages 52–71, Berlin, Heidelberg, 1981. Springer Berlin Heidelberg.
- [Doreian *et al.*, 1996] Patrick Doreian, Roman Kapuscinski, David Krackhardt, and Janusz Szczypula. A brief history of balance through time. *The Journal of Mathematical Sociology*, 21(1-2):113–131, 1996.
- [Emerson and Clarke, 1982] E.Allen Emerson and Edmund M. Clarke. Using branching time temporal logic to synthesize synchronization skeletons. *Science of Computer Programming*, 2(3):241–266, 1982.
- [Harary *et al.*, 1965] Frank Harary, Robert Z. Norman, and Dorwin Cartwright. *Structural Models: An Introduction to the Theory of Directed Graphs*. John Wiley & Sons Inc, New York, September 1965.
- [Harary, 1953] Frank Harary. On the notion of balance of a signed graph. *Michigan Mathematical Journal*, 2(2):143–146, 1953.
- [Heider, 1944] Fritz Heider. Social perception and phenomenal causality. *Psychological Review*, 51(6):358–374, 1944.
- [Heider, 1946] Fritz Heider. Attitudes and cognitive organization. *Journal of Psychology*, 21(1):107–112, 1946.
- [Heider, 1958] Fritz Heider. *The Psychology of Interpersonal Relations*. John Wiley & Sons, 1958.
- [Hummon and Doreian, 2003] Norman P Hummon and Patrick Doreian. Some dynamics of social balance processes: bringing Heider back into balance theory. *Social Networks*, 25(1):17–49, 2003.
- [Kulakowski, 2007] Krzysztof Kulakowski. Some recent attempts to simulate the Heider balance problem. *Computing in Science and Engineering*, 9:80–85, 06 2007.
- [Newcomb, 1961] Theodore M. Newcomb. *Acquaintance Process*. Holt, Rinehart & Winston, New York, December 1961.
- [Radicchi *et al.*, 2007] Filippo Radicchi, Daniele Vilone, Sooyeon Yoon, and Hildegard Meyer-Ortmanns. Social balance as a satisfiability problem of computer science. *Physical Review E*, 75:026106, Feb 2007.

- [Stockmeyer and Meyer, 1973] L. J. Stockmeyer and A. R. Meyer. Word problems requiring exponential time (preliminary report). In *Proceedings of the Fifth Annual ACM Symposium on Theory of Computing*, pages 1–9, 1973.
- [Wang and Thorngate, 2003] Zhigang Wang and Warren Thorngate. Sentiment and social mitosis: Implications of Heider’s balance theory. *Journal of Artificial Societies and Social Simulation*, 6(3):12–23, June 2003.
- [Zheng *et al.*, 2015] Xiaolong Zheng, Daniel Zeng, and Fei-Yue Wang. Social balance in signed networks. *Information Systems Frontiers*, 17(5):1077–1095, January 2015.